基于属性优化矩阵补全的抗托攻击推荐算法 *

周宇轩[a]，陈　蕾[a, b]，张涵峰[a]

(南京邮电大学 a. 计算机学院; b. 江苏省无线传感网高技术研究重点实验室，南京 210003)

摘　要：托攻击是当前推荐系统面临的严峻挑战之一。由于推荐系统的开放性，恶意用户可轻易对其注入精心设计的评分从而影响推荐结果，降低用户体验。基于属性优化结构化噪声矩阵补全技术，提出一种鲁棒的抗托攻击个性化推荐（SATPR）算法，将攻击评分视为评分矩阵中的结构化行噪声并采用 $L_{2,1}$ 范数进行噪声建模，同时引入用户与物品的属性特征以提高托攻击检测精度。实验表明，SATPR 算法在托攻击下可取得比传统推荐算法更精确的个性化评分预测效果。

关键词：推荐系统；托攻击；$L_{2,1}$ 范数正则化；属性特征

中图分类号：TP301.6　　doi: 10.3969/j.issn.1001-3695.2017.09.0906

Shilling-attack-tolerant recommendation algorithm based on attribute facilitated matrix completion

Zhou Yuxuan[a], Chen Lei[a, b], Zhang Hanfeng[a]

(*a. School of Computer Science*, *b. Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks*, *Nanjing University of Posts & Telecommunications*, *Nanjing* 210003, *China*)

Abstract: Shilling attack is one of serious challenges which recommender systems are facing. Malicious users can easily insert well-designed ratings into recommender systems to affect recommendation results and decrease user experiences because of the openness of recommender systems. This article proposed a robust shilling-attack-tolerant personalized recommendation (SATPR) algorithm based on attribution facilitated matrix completion with structural noise technology, regarded the ratings of attack users in the rating matrix as structural row noise and modeled them with $L_{2,1}$-norm. This article also introduced attributive characters of users and items to improve the accuracy of detection of shilling-attack. Experimental results showed that SATPR algorithm achieved more accurate results of personalized rating prediction than traditional recommendation algorithms under shilling attacks.

Key Words: recommender system; shilling attack; $L_{2,1}$-norm regularization; attributive characters

0　引言

面对信息过载问题，推荐系统应运而生。推荐系统是一种软件系统，它通过收集用户信息，物品信息以及用户与物品的交互信息，了解用户的偏好，从而将用户可能感兴趣的物品推荐给用户，在一定程度上解决困扰用户的信息过载问题[1]。当前实现推荐系统的一种主流方法是协同过滤（collaborative filtering）方法。它依赖于用户的历史行为，分析过去的用户-物品交互，建立新的用户-物品联系[2]。协同过滤推荐算法的推荐结果与所收集到的用户-物品交互历史记录息息相关，当收集到的用户-物品交互数据越能精确反映出用户偏好与物品本质，最终的推荐结果也将越精确。然而，用户-物品交互数据的产生者是所有用户，并没有设置准入门槛，这种数据来源的开放性导致了协同过滤推荐系统极易受到恶意用户的干扰，这种现象称为托攻击(shilling attack)。攻击者往往可以轻易伪装成普通用户，在与物品交互的过程中，有目的性地进行操作，从而干扰了正常的推荐结果，严重影响了用户体验。因此，抵御托攻击的干扰，增强推荐系统的鲁棒性，成为推荐系统的重要改进方向。

通常情况下，试图区分精心伪装的托攻击用户与正常用户是很困难的。然而，从攻击意图，攻击过程等各方面分析，托攻击概貌与正常用户概貌间依然存在着细微差别。本文针对用户-物品评分矩阵中托攻击概貌所反映出的用户偏好不够自然，异于正常用户偏好这一性质，将评分矩阵中的托攻击概貌建模为影响评分矩阵低秩性的结构化行噪声，并采用矩阵$L_{2,1}$范数对

基金项目：江苏省自然科学基金面上项目（BK20161516）；中国博士后科学基金资助项目（2015M581794）；江苏省高校自然科学研究面上项目（15KJB520027）；江苏省博士后科研资助计划资助项目（1501023C）

作者简介：周宇轩（1991-），男，硕士研究生，主要研究方向为推荐系统、机器学习（1015041123@njupt.edu.cn）；陈蕾（1975-），男，副教授，硕导，博士，主要研究方向为神经网络、模式识别、机器学习及其在医学影像分析中的应用；张涵峰（1996-），男，本科生，主要研究方向为推荐系统.

其进行解析。除了考虑用户概貌的差别，本文也将用户与物品属性信息纳入考虑之中，进一步提高了检测精度。当检测出托攻击用户之后，剔除相应攻击评分再进行评分预测，将显著增强评分预测的鲁棒性。基于以上工作，本文提出了一种基于属性优化矩阵补全的抗托攻击个性化推荐算法（SATPR）。

1　相关工作

托攻击是当前推荐系统遇到的严峻挑战之一。恶意商家或用户为了达成其特殊目的（往往是经济利益），冒充正常用户，在与物品交互的过程中，向推荐系统注入精心设计的虚假用户概貌，从而干扰正常的推荐结果，实现其攻击意图。在一个攻击用户概貌中，所有物品的集合I可被分成四个子集：选择填充项（selected items）I^S，填充项（filler items）I^F，目标项（target items）I^T和未评分项（unrated items）$I^\emptyset$。选择填充项中的物品由攻击者刻意挑选，用以接近特定目标用户群体；填充项中的物品通常是随机挑选的，用以隐藏攻击者的身份；目标项即被攻击的目标物品；另外，由于实际用户不可能对所有物品进行评价，这些没有获得评分的物品构成未评分项。根据攻击意图，托攻击分为推攻击(push attack)与核攻击(nuke attack)[3]。一些攻击者致力于提升目标物品的知名度，从而使推荐系统将其作为推荐结果推送给用户，这种攻击称为推攻击；另一些攻击者则致力于降低目标物品的知名度，从而使该目标物品被推荐的可能性降低，这种攻击称为核攻击。而根据实施攻击的具体细节，托攻击可分为多种攻击类型，其中最常见的三种攻击模型分别是随机攻击（random attack），均值攻击（average attack）和流行攻击（bandwagon attack），各攻击模型之间的区别主要在于攻击用户概貌所采用的构造规则不同。在随机攻击中，攻击概貌没有选择填充项；填充项随机选择物品，并根据全体物品平均分为其打分；对于目标项，根据推攻击或核攻击，分别为其打高分或低分。在均值攻击中，攻击概貌同样没有选择填充项；填充项随机选择物品，其分值不再根据全体物品平均分，而是根据对应物品各自平均分进行打分；对于目标项，根据推攻击或核攻击，分别为其打高分或低分。在流行攻击的推攻击中，选择填充项选择最流行的物品打高分；填充项随机选择物品，并根据全体物品平均分为其打分；对目标项进行推攻击打高分。流行攻击的核攻击形式也称为反流行攻击，其中选择填充项选择最不受欢迎的物品打低分，填充项同样随机选择物品，并根据全体物品平均分为其打分；对目标项进行核攻击打低分。

由此可见，托攻击用户概貌通常进行了精心设计与伪装，攻击概貌的注入将极大改变原本正常的推荐结果。因此，托攻击问题引起了推荐系统领域内许多研究者的关注。Zhang 等人[4]基于用户概貌偏离度，结合了用户可信度与物品可信度，以可信度加权提出了一种分类属性，称为信任加权的平均偏离度（trust-weighted deviation from mean agreement，TWDMA），用以区分攻击用户与正常用户；Bryan 等人[5]利用 Hv-score 度量对用户进行排序，由排序后的用户概貌确定攻击目标项，在获知攻击目标项的基础上以滑动窗的形式对用户概貌进行排查；Li 等人[6]根据攻击概貌中物品的选择是基于人为制造的规则，有悖于自然状态下正常用户的内在偏好这一性质，通过挖掘用户概貌中的物品属性特征评判用户概貌；Deng 等人[7]提出了一种结合了主成分分析(PCA)与扰动的无监督检测方法，对用户概貌加入高斯噪声，在加入噪声前后各做一次 PCA，结合两次的结果进行比较分析，确定托攻击用户。

另外，由于推荐系统往往要针对用户-物品评分矩阵进行研究，因此，在矩阵补全领域对于异常元素的范数正则化技术同样值得借鉴。陈蕾等人[8]对于低秩矩阵补全在现实应用中遇到的采样数据可能会受到结构化噪声污染的问题，提出了一种利用$L_{2,1}$范数解析结构化噪声的低秩矩阵补全算法；Xiao 等人[9]在低秩矩阵补全问题中，对于采样过程中高斯噪声在矩阵补全问题中产生的干扰，使用了矩阵的 Frobenius 范数来解析高斯噪声，提高了算法的鲁棒性；Zhang 等人[10]在 2D 图像特征提取时利用$L_{2,1}$范数有效缓解了算法对于噪声敏感的问题；汤镇宇等人[11]在基于字典求取稀疏系数以进行人脸识别的过程中利用了L_2范数正则化，使算法在降低复杂度的同时保持了较高的识别精度。

2　基于属性优化结构化噪声矩阵补全的抗托攻击个性化推荐算法（SATPR）

2.1　属性优化结构化噪声矩阵补全模型（AFMCSN）

通常在推荐系统中，m 名用户构成用户集合：$U=\{u_1,u_2,\cdots u_m\}$，同样的，n 件物品构成物品集合：$I=\{i_1,i_2,\cdots i_n\}$，用户-物品评分矩阵可表示为$R\in\mathbb{R}^{m\times n}$：

$$R=\begin{bmatrix} ? & ? & ? \\ * & r_{i,j} & * \\ * & ? & ? \end{bmatrix} \tag{1}$$

其中：用户 i 对物品 j 的评分记作$r_{i,j}$，“*”表示已知评分，“？”表示未知评分。由于每个用户只可能对有限的物品进行评分，而每件物品也仅可能收到有限用户的评分，故这个评分矩阵通常包含大量的空缺评分，是一个稀疏矩阵。推荐系统的目的就是预测用户对于未评分物品的偏好程度，从而将合适的物品推荐给相应用户。

在推荐系统中，大量的用户之间和大量的物品之间必然存在着偏好相近的用户和属性相近的物品，这种相近性质使得用户-物品评分矩阵往往具有近似低秩性[12]，故推荐系统问题可利用低秩矩阵补全技术进行评分预测。用 $\boldsymbol{R}$ 表示当前观察到的评分矩阵，则推荐系统问题可建模为

$$\min_{X\in\mathbb{R}^{m\times n}} rank(X) \quad s.t. P_\Omega(R)=P_\Omega(X) \tag{2}$$

其中:Ω集合表示评分矩阵中已收到评分的元素下标集合，$\Omega\subseteq\{1,2\cdots m\}\times\{1,2\cdots n\}$。$P_\Omega(\cdot)$是投影算子，表示当元素下标$(i,j)\in\Omega$时，得到对应位置采样元素：

$$[\mathrm{P}_\Omega(\boldsymbol{R})]_{i,j}=\begin{cases}r_{i,j}(i,j)\in\Omega\\0(i,j)\notin\Omega\end{cases}\tag{3}$$

但是，秩函数rank($\boldsymbol{X}$)是非凸的，直接使用秩函数建模得到的是一个 NP-Hard 问题，其计算代价会随着问题规模的扩大而急剧增大。因此，往往将秩函数松弛化为核范数来解决此问题[13]：

$$\min_{\boldsymbol{X}\in\mathbb{R}^{m\times n}}\|\boldsymbol{X}\|_*\,\mathrm{P}_\Omega(\boldsymbol{R})=\mathrm{P}_\Omega(\boldsymbol{X})\tag{4}$$

其中:$\|\boldsymbol{X}\|_*=\sum_{i=1}^{\min(m,n)}\sigma_i$为矩阵核范数，$\sigma_i$为矩阵 **X** 的第 i 大奇异值。

在实际应用中，推荐系统往往会遭受恶意用户的托攻击。面对托攻击干扰，标准矩阵补全模型的推荐精度将严重降低。因此，为了保证推荐的质量，有必要抵御这些攻击数据的影响。通过对托攻击特点进行分析可知，托攻击的目的是更改原来自然情况下的评分情况，假如在自然情况下的评分情况已经符合攻击者的意图，则没有进行攻击的必要，故托攻击用户的评分与自然状态下正常用户的评分存在不一致性。另外，托攻击用户的评分通常是机械填充的，这也会与基于兴趣偏好而形成的正常用户评分具有相异之处。基于以上分析，本文将用户-物品评分矩阵中的托攻击评分建模为结构化行噪声，这些结构化噪声与正常用户评分的潜在规律相违背，打破了评分矩阵的近似低秩性。对于评分矩阵中存在的结构化行噪声，可利用矩阵$L_{2,1}$范数对其进行解析[14]。在剔除攻击评分之后，再进行评分预测，将有效提高推荐精度。此时，模型可改进为

$$\min_{\boldsymbol{X},\boldsymbol{Z}\in\mathbb{R}^{m\times n}}\|\boldsymbol{X}\|_*+\lambda\|\boldsymbol{Z}\|_{2,1}\quad \mathrm{s.t.}\,\mathrm{P}_\Omega(\boldsymbol{R})=\mathrm{P}_\Omega(\boldsymbol{X}+\boldsymbol{Z})\tag{5}$$

其中：$\|\boldsymbol{Z}\|_{2,1}=\sum_{i=1}^{m}\sqrt{\sum_{j=1}^{n}z_{i,j}^2}$为矩阵 $L_{2,1}$ 范数。

此外，用户往往会出于情绪波动而打出不太精确的评分，为了平滑评分的这种细微波动，可引入矩阵的 Frobenius 范数，将其改写成罚函数形式：

$$\min_{\boldsymbol{X},\boldsymbol{Z}\in\mathbb{R}^{m\times n}}\|\boldsymbol{X}\|_*+\lambda\|\boldsymbol{Z}\|_{2,1}+\frac{\beta}{2}\|\mathrm{P}_\Omega(\boldsymbol{X}+\boldsymbol{Z}-\boldsymbol{R})\|_F^2\tag{6}$$

其中：$\|\boldsymbol{X}\|_F=\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|x_{i,j}|^2}$为矩阵的 Frobenius 范数。

正如之前所叙述的，评分矩阵通常是稀疏矩阵，能够收到的评分数远小于评分矩阵元素个数。在 MovieLens 的 ml-20m 数据集中，138493 名用户对于 27278 部电影仅有 20000263 个评分；在 EachMovie 数据集中，72916 个用户对 l628 部电影仅进行了 2811983 次评分。基于稀疏数据求取高维未知矩阵很难确保求解的准确性。为解决数据稀疏性问题[15]，本文考虑引入属性特征信息，将简单的评分矩阵 $\boldsymbol{R}$ 细化为三个矩阵的乘积$\boldsymbol{R}=\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{B}$，其中，$\boldsymbol{A}$ 和 $\boldsymbol{B}$ 分别是用户特征矩阵和物品特征矩阵，$\boldsymbol{A}\in\mathbb{R}^{k\times m}$，$\boldsymbol{B}\in\mathbb{R}^{k\times n}$，矩阵列向量分别是用户特征向量和物品特征向量，即量化了的属性信息；$\boldsymbol{X}$ 矩阵是低维未知矩阵，$\boldsymbol{X}\in\mathbb{R}^{k\times k}$。此时，模型可改为

$$\min_{\boldsymbol{X}\in\mathbb{R}^{k\times k},\boldsymbol{Z}\in\mathbb{R}^{m\times n}}\|\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{B}\|_*+\lambda\|\boldsymbol{Z}\|_{2,1}+\frac{\beta}{2}\|\mathrm{P}_\Omega(\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{B}+\boldsymbol{Z}-\boldsymbol{R})\|_F^2\tag{7}$$

然而，由于rank($\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{B}$) ≤ min{rank($\boldsymbol{A}$)，rank($\boldsymbol{B}$)}，即只要特征矩阵 $\boldsymbol{A}$ 与 $\boldsymbol{B}$ 其中之一的秩足够小，比如 $\boldsymbol{A}$ 或 $\boldsymbol{B}$ 的维度过小，包含信息过少等，则无论 $\boldsymbol{X}$ 矩阵取何值，乘积$\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{B}$的秩都必然会更小，将直接满足低秩条件。为避免这种解的任意性，本文仅对待求变量 $\boldsymbol{X}$ 进行低秩约束。综上所述，属性优化结构化噪声矩阵补全（Attribution Facilitated Matrix Completion with Structural Noise，AFMCSN）模型可建模为

$$\min_{\boldsymbol{X}\in\mathbb{R}^{k\times k},\boldsymbol{Z}\in\mathbb{R}^{m\times n}}\|\boldsymbol{X}\|_*+\lambda\|\boldsymbol{Z}\|_{2,1}+\frac{\beta}{2}\|\mathrm{P}_\Omega(\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{B}+\boldsymbol{Z}-\boldsymbol{R})\|_F^2\tag{8}$$

当求得行噪声矩阵 $\boldsymbol{Z}$ 之后，便可据此排除托攻击干扰，获得高精度的评分预测结果。

2.2 AFMCSN 模型的优化求解

为求解 AFMCSN 模型，可采用分块坐标下降算法[16]，通过对各个变量交替最小化得到最优解。为了进一步方便求解，本文引入变量 $\boldsymbol{C}$ 并令$\boldsymbol{C}=\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{B}$，模型（8）转换为

$$\min_{\boldsymbol{X}\in\mathbb{R}^{k\times k},\boldsymbol{C},\boldsymbol{Z}\in\mathbb{R}^{m\times n}}\|\boldsymbol{X}\|_*+\lambda\|\boldsymbol{Z}\|_{2,1}+\frac{\beta}{2}\|\mathrm{P}_\Omega(\boldsymbol{C}+\boldsymbol{Z}-\boldsymbol{R})\|_F^2$$
$$s.t.\ \boldsymbol{C}-\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{B}=0\tag{9}$$

同上，利用 Frobenius 范数将其改写成罚函数形式：

$$\min_{\boldsymbol{X}\in\mathbb{R}^{k\times k},\boldsymbol{C},\boldsymbol{Z}\in\mathbb{R}^{m\times n}}\|\boldsymbol{X}\|_*+\lambda\|\boldsymbol{Z}\|_{2,1}+\frac{\beta}{2}\|\mathrm{P}_\Omega(\boldsymbol{C}+\boldsymbol{Z}-\boldsymbol{R})\|_F^2+\frac{\rho}{2}\|\boldsymbol{C}-\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{B}\|_F^2\tag{10}$$

不妨令

$$\mathrm{L}_\rho(\boldsymbol{X},\boldsymbol{Z},\boldsymbol{C})=\|\boldsymbol{X}\|_*+\lambda\|\boldsymbol{Z}\|_{2,1}+\frac{\beta}{2}\|\mathrm{P}_\Omega(\boldsymbol{C}+\boldsymbol{Z}-\boldsymbol{R})\|_F^2+\frac{\rho}{2}\|\boldsymbol{C}-\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{B}\|_F^2\tag{11}$$

对各个变量交替最小化，则各个变量的迭代更新公式为

$$\begin{cases}\boldsymbol{X}^{k+1}=\arg\min\limits_{\boldsymbol{X}\in\mathbb{R}^{k\times k}}\mathrm{L}_\rho(\boldsymbol{X},\boldsymbol{Z}^k,\boldsymbol{C}^k)\\\boldsymbol{Z}^{k+1}=\arg\min\limits_{\boldsymbol{Z}\in\mathbb{R}^{m\times n}}\mathrm{L}_\rho(\boldsymbol{X}^{k+1},\boldsymbol{Z},\boldsymbol{C}^k)\\\boldsymbol{C}^{k+1}=\arg\min\limits_{\boldsymbol{C}\in\mathbb{R}^{m\times n}}\mathrm{L}_\rho(\boldsymbol{X}^{k+1},\boldsymbol{Z}^{k+1},\boldsymbol{C})\end{cases}\tag{12}$$

至此，原最优化问题变为三个子问题，分别对应$\boldsymbol{X},\boldsymbol{Z},\boldsymbol{C}$的更新。

对子问题 1，本文采用近邻前向后向分裂（proximal forward backward splitting, PFBS）[17] 技术对其进行优化求解，令:

$$\begin{cases}\mathrm{F}_1(\boldsymbol{X})=\|\boldsymbol{X}\|_*\\\mathrm{F}_2(\boldsymbol{X})=\frac{\rho}{2}\|\boldsymbol{C}-\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{B}\|_F^2\end{cases}\tag{13}$$

其中函数$\mathrm{F}_2(\boldsymbol{X})$的导数为

$$\nabla\mathrm{F}_2(\boldsymbol{X})=\rho(\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{B}\boldsymbol{B}^T-\boldsymbol{A}\boldsymbol{C}\boldsymbol{B}^T)\tag{14}$$

为简化公式，方便求解，引入一个新变量 $\boldsymbol{Y}$，令

$$\boldsymbol{Y}^{k+1}=\boldsymbol{X}^k-\delta_x\nabla\mathrm{F}_2(\boldsymbol{X}^k)=\boldsymbol{X}^k-\delta_x\rho(\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{B}\boldsymbol{B}^T-\boldsymbol{A}\boldsymbol{C}^k\boldsymbol{B}^T)\tag{15}$$

其中，根据 PFBS 规则，引入参数δ_x用于对 $\boldsymbol{X}$ 进行迭代更新。则

$$\boldsymbol{X}^{k+1}=\arg\min_{\boldsymbol{X}\in\mathbb{R}^{k\times k}}\frac{1}{2}\|\boldsymbol{X}-\boldsymbol{Y}^{k+1}\|_F^2+\delta_x\|\boldsymbol{X}\|_*\tag{16}$$

根据文献[8]，对于矩阵$\boldsymbol{Y}\in\mathbb{R}^{k\times k}$和常数$\tau>0$，有

$$\mathrm{D}_\tau(\boldsymbol{Y})=\arg\min_{\boldsymbol{X}\in\mathbb{R}^{k\times k}}\left\{\frac{1}{2}\|\boldsymbol{X}-\boldsymbol{Y}\|_F^2+\tau\|\boldsymbol{X}\|_*\right\}\tag{17}$$

其中：$\mathrm{D}_\tau(\boldsymbol{Y})$是奇异值阈值算子，若矩阵 $\boldsymbol{Y}$ 的奇异值分解为$\boldsymbol{Y}=$

$U\Sigma V^T$，则τ所对应的奇异值阈值算子为$D_\tau(Y)=U*[\text{sign}(Y)\circ \max(|Y|-\tau,0)]*V^T$，其中符号“∘”是 Hadamard 积，表示两矩阵对应元素相乘。

因此，X的更新可按如下步骤迭代进行：

$$\begin{cases}Y^{k+1}=(X^k-\delta_X\rho AA^TXBB^T-AC^kB^T)\\X^{k+1}=D_{\delta_X}(Y^{k+1})\end{cases}\tag{18}$$

在 PFBS 中，参数δ需满足$0<\delta<\frac{2}{L_f}$。通过计算，$L_f_X=\sigma_{\max}(BB^T)*\sigma_{\max}(AA^T)$，故参数$\delta_X$需满足$0<\delta_X<\frac{2}{\sigma_{\max}(BB^T)*\sigma_{\max}(AA^T)}$，在实验中，本文取$\delta_X=\frac{1}{\sigma_{\max}(BB^T)*\sigma_{\max}(AA^T)}$。

对于子问题 2，类似于X的更新过程，令

$$F_2(Z)=\frac{\beta}{2}\|P_\Omega(C+Z-R)\|_F^2\tag{19}$$

其导数为

$$\nabla F_2(Z)=\beta P_\Omega(C+Z-R)\tag{20}$$

为简化公式，方便求解，引入一个新变量V，令

$$V^{k+1}=Z^k-\delta_Z\nabla F_2(Z^k)=Z^k-\delta_Z\beta P_\Omega(C^k+Z^k-R)\tag{21}$$

则

$$Z^{k+1}=\arg\min_{Z\in\mathbb{R}^{m\times n}}\frac{1}{2}\|Z-V^{k+1}\|_F^2+\delta_Z\lambda\|Z\|_{2,1}\tag{22}$$

由文献[8]，对于Z矩阵内每一行：

$$(Z^{k+1})^{(i)}=\max\left\{1-\frac{\delta_Z\lambda}{\left\|(V^{k+1})^{(i)}\right\|_2},0\right\}*(V^{k+1})^{(i)}$$
$$i=1,2,\cdots,m\tag{23}$$

因此，Z的更新可按照如下步骤迭代进行：

$$\begin{cases}V^{k+1}=Z^k-\delta_Z\beta P_\Omega(C^k+Z^k-R)\\(Z^{k+1})^{(i)}=\max\left\{1-\frac{\delta_Z\lambda}{\left\|(V^{k+1})^{(i)}\right\|_2},0\right\}*(V^{k+1})^{(i)}\quad i=1,2,\cdots,m\end{cases}\tag{24}$$

类似地，通过计算，$L_f_Z=\beta$，故参数δ_Z需满足$0<\delta_Z<\frac{2}{\beta}$。，在实验中，取$\delta_Z=\frac{1}{\beta}$。

对于子问题 3，令

$$F(C)=\frac{\beta}{2}\|P_\Omega(C+Z-R)\|_F^2+\frac{\rho}{2}\|C-A^TXB\|_F^2\tag{25}$$

即

$$C^{k+1}=\arg\min_{C\in\mathbb{R}^{m\times n}}F(C)\tag{26}$$

此时有

$$\nabla F(C)=\beta P_\Omega(C+Z-R)+\rho(C-A^TXB)\tag{27}$$

令$\nabla F(C)=0$，可求得C的迭代更新公式：

$$C^{k+1}=[\rho A^TX^{k+1}B+\beta P_\Omega(R-Z^{k+1})]/(\rho+\beta)\tag{28}$$

至此，可整理得到 AFMCSN 模型的求解步骤，如算法 1 所示。

算法 1 属性优化结构化噪声矩阵补全算法（AFMCSN）

输入：用户与物品特征矩阵A，B，采样评分矩阵R，采样下标集合Ω，参数β,ρ,λ以及迭代次数 *Maxiter*

输出：结构化行噪声矩阵Z

```
1  initialize  X^0 = 0, Z^0 = 0, C^0 = 0;
2  for  k=0  to  Maxiter
3        根据式(18)更新 X;
4        根据式(24)更新 Z;
5        根据式(28)更新 C;
6  end  for
7  return  Z;
```

其中，最主要的算法步骤集中在 3~5 步，即按照式(18)(24)(28)更新各个变量。在用户数为m，物品数为n，属性特征向量长度为k的情况下，分别考虑式(18)(24)(28)的时间复杂度：式(18)包括中间变量Y的更新以及X的更新。在更新Y的过程中，需要进行矩阵连续相乘的操作，其时间复杂度为$O(mnk)$；对于X的更新，需要对矩阵Y进行奇异值分解，其时间复杂度为$O(k^3)$；式(24)包括中间变量V的更新以及Z的更新。在更新V的过程中，主要是矩阵加减操作，其时间复杂度为$O(mn)$；对于Z的更新，需要逐行更新，每一行需要进行n次相乘，其时间复杂度为$O(mn)$；式(28)对应C的更新，其中主要的计算步骤在于矩阵相乘部分，其时间复杂度为$O(mnk)$。

综上，算法 1 的总时间复杂度为$O(Maxiter*(mnk+mn+k^3))$，其中计算代价较大的部分在于矩阵相乘操作与奇异值分解操作。对于矩阵相乘，当前学术界与产业界已经提出并实现了很多高效的优化算法，以本实验所采用的 MATLAB 平台为例，它在执行矩阵运算时可使用 Intel 的 Math kernel library 库(https://software.intel.com/en-us/mkl)，从底层硬件角度进行了优化，运算速度可得到极大提高。而矩阵的奇异值分解可利用 PROPACK 软件包(http://sun.stanford.edu/~rmunk/PROPACK/)，首先估计矩阵的秩r，计算过程中仅对前r个最大奇异值以及对应奇异向量进行计算，即将时间复杂度从$O(k^3)$降为$O(rk^2)$，能够在降低运算代价的同时保证运算精度。

2.3 SATPR 算法

基于 AFMCSN 托攻击检测模型，本文可得到抗托攻击的 SATPR 算法。首先，SATPR 算法需要引入代表用户与物品属性特征的A矩阵和B矩阵，如果数据集中已经提供了这两个属性特征矩阵，便可直接利用；但更常见的情形是，数据集中并未直接提供这两个特征矩阵。此时需要先计算用户间相似度与物品间相似度，再利用谱聚类算法分别构建用户与物品的属性特征矩阵[18]。在推荐系统领域，相似度是用来预测用户评分的常用工具，多种相似度算法均可应用于属性特征矩阵的构建。在此，引入 Shi 等人[19]提出的 HeteSim 相似度,这是一个递归定义的相似度度量，基于对象的属性信息考虑了两个对象之间所有可能产生关联的元路径，以入度与出度数量衡量彼此之间的影响程度，可以细致地表现出对象之间的相似性。

当获取了相似度信息，便可利用谱聚类算法，基于相似度计算属性特征矩阵。首先计算拉普拉斯矩阵的特征值和特征向量，取前k个特征向量组成矩阵，每一行即可视为对应样本的属性特征向量，这些属性特征向量之间的相似关系可以恰当反映对应的对象之间的潜在相似程度。通过这种方法，同样可以获得代表属性特征的矩阵A和B。

具备了必要条件后，便可利用 AFMCSN 模型求解结构化

行噪声矩阵 $\boldsymbol{Z}$，然后根据 $\boldsymbol{Z}$ 矩阵剔除攻击项，再利用传统推荐算法进行评分预测。当前，针对无托攻击评分数据集已经有了很多行之有效的高精度推荐算法，但在托攻击数据集上，这些传统推荐算法往往会面临性能下降的问题。而将传统推荐算法结合本文所提出的 AFMCSN 托攻击检测模型，在托攻击污染的数据集上将同样可保证评分预测的鲁棒性。在此本文采用 Shi Chuan[20]等提出的 SimMF 算法进行评分预测，这种算法基于矩阵分解，同时又充分利用了相似度信息进行加权，取得了极高的预测精度。

至此，可整理得到 SATPR 算法步骤，如算法 2 所示。

算法 2 基于属性优化的抗托攻击个性化推荐算法（SATPR）

输入：用户与物品属性信息，采样评分矩阵 $\boldsymbol{R}$，采样下标集合Ω，参数β,ρ,λ以及迭代次数 Maxiter。

输出：预测评分矩阵$\hat{\boldsymbol{R}}$。

1 基于用户与物品属性信息计算用户相似度与物品相似度;

2 基于相似度，利用谱聚类算法分别构建用户与物品的属性特征矩阵 $\boldsymbol{A}$，$\boldsymbol{B}$;

3 基于属性特征矩阵 $\boldsymbol{A}$，$\boldsymbol{B}$，利用算法 1 求解结构化行噪声矩阵 $\boldsymbol{Z}$;

4 根据矩阵 $\boldsymbol{Z}$ 从采样评分矩阵剔除托攻击评分;

5 利用传统推荐算法进行评分预测，求得预测评分矩阵 $\hat{\boldsymbol{R}}$;

3 实验与分析

3.1 数据集

本文实验所采用的数据集来自于真实的 Douban Book 数据，包含了 1000 名用户对于 2000 本书的 32460 个评分。值得注意的是，在这个数据集中不仅有用户对于书籍的评分数据，还包括用户的属性信息以及书籍的属性信息。对于书籍来说，其属性信息包括四类：作者，书名，出版社，出版时间；对于用户来说，其属性信息包括三类：群组，位置，好友。由于属性个数相对过少，将通过谱聚类技术构建代表用户与书籍属性特征的 $\boldsymbol{A}$ 矩阵和 $\boldsymbol{B}$ 矩阵。实验假定数据集中原有的用户均为正常用户，在原评分矩阵中随机选取用户评分行，利用托攻击模型生成托攻击概貌对其进行替换，得到托攻击干扰下的评分数据作为实验数据集。

3.2 评价指标

从托攻击用户检测的角度来看，对于托攻击用户的判断可视为一种分类问题，对此，常用的度量标准有查准率（precision），查全率（recall）和 F1 度量。

$$\text{precision}=\frac{TP}{TP+FP} \tag{29}$$

$$\text{recall}=\frac{TP}{TP+FN} \tag{30}$$

$$F1=\frac{2\times\text{precision}\times\text{recall}}{\text{precision+recall}} \tag{31}$$

其中：TP 表示判断为攻击用户的集合中，真正的攻击用户数量；FP 表示判断为攻击用户的集合中，被误判的正常用户数量；FN 表示判断为正常用户的集合中，隐藏的攻击用户数量。precision 越高说明所找出的攻击用户确实是真正的攻击用户的可能性越大，recall 越高说明真正的攻击用户被检测出的可能性越大。F1 度量是一个更为平衡的选择，兼顾了 precision 与 recall 度量，F1 度量越大，说明检测效果越精确。

由于推荐系统问题可以视为对用户-物品评分矩阵的缺失评分预测问题，因此从矩阵补全的角度，可以选用平均绝对误差(mean absolute error，MAE)和均方根误差(root mean square error，RMSE)考察评分预测的精度：

$$\text{MAE}=\frac{\sum_{(i,j)\in T}\left|\widehat{r_{i,j}}-r_{i,j}\right|}{|T|} \tag{32}$$

$$\text{RMSE}=\sqrt{\frac{\sum_{(i,j)\in T}\left(\widehat{r_{i,j}}-r_{i,j}\right)^2}{|T|}} \tag{33}$$

其中：T表示测试集元素下标集合，$\widehat{r_{i,j}}$表示用户 i 对物品 j 的预测评分，$r_{i,j}$表示用户 i 对物品 j 的实际评分。MAE 与 RMSE 的值越小，则推荐算法对评分矩阵中缺失评分的预测精度越高。

3.3 实验设计与分析

本节将通过实验检验 SATPR 算法在托攻击下的性能。实验中对评分矩阵分别施加随机攻击，均值攻击和流行攻击三种最常见的攻击方式，各自进行推攻击与核攻击，产生六种不同的攻击类型。在各攻击类型下分别施加填充率为 10%，20%，攻击强度为 5%，10%的一系列托攻击，并在这些托攻击干扰后的评分矩阵上进行实验。这里的填充率和攻击强度是描述攻击规模常用的两个统计学指标[21]，填充率p^{fill}通常用于描述攻击概貌的结构，定义为

$$p^{fill}=|\{填充物品\}|/n \tag{34}$$

其中：{填充物品}表示攻击概貌中除了攻击目标之外所填充的其余物品构成的集合，n 表示物品总数。而攻击强度p^{att}用于描述攻击用户存在的比重，定义为

$$p^{att}=|\{攻击用户\}|/|\{正常用户\}| \tag{35}$$

其中{攻击用户}表示托攻击用户构成的集合，{正常用户}表示正常用户构成的集合，$|\cdot|$是集合的势，在此处表示集合中元素个数。

在算法中所涉及的参数主要有β,ρ,λ，其取值可以通过在训练集上采用交叉验证的方式获得。本文设计了两组实验。第一组实验从托攻击用户查找的角度出发，检验 SATPR 算法在各种攻击下对于攻击用户的检测能力；第二组实验从评分预测的角度进行考虑，检验 SATPR 算法在各种攻击模式下的评分预测精度。

实验 1 SATPR 算法检测实验

在本实验中，将考察各种攻击下 SATPR 算法对于托攻击用户的查准率，查全率以及 F1 度量。作为对比，本实验分别考虑了加入属性信息前与加入属性信息后，算法对于托攻击用户

的检测性能。对于加入属性信息前的算法，将其记作SATPR_NoAttr。实验结果如表 1~6 所示。

表 1 随机攻击下 SATPR_NoAttr 算法的查准率,查全率和 F1 度量

	p^{att}	p^{fill}	precision	recall	F1
Push	10%	5%	0.7222	1.0000	0.8387
	10%	10%	0.8667	1.0000	0.9286
	20%	5%	0.7979	0.9222	0.8556
	20%	10%	0.8107	1.0000	0.8954
Nuke	10%	5%	0.6947	1.0000	0.8198
	10%	10%	0.8750	1.0000	0.9333
	20%	5%	0.7455	1.0000	0.8542
	20%	10%	0.8068	1.0000	0.8930

表 2 随机攻击下 SATPR 算法的查准率,查全率和 F1 度量

	p^{att}	p^{fill}	precision	recall	F1
Push	10%	5%	0.7521	1.0000	0.8585
	10%	10%	0.9891	1.0000	0.9945
	20%	5%	0.8564	1.0000	0.9227
	20%	10%	0.9824	1.0000	0.9911
Nuke	10%	5%	0.7647	1.0000	0.8667
	10%	10%	0.9785	1.0000	0.9891
	20%	5%	0.8392	1.0000	0.9126
	20%	10%	0.9824	1.0000	0.9911

表 3 均值攻击下 SATPR_NoAttr 算法的查准率,查全率和 F1 度量

	p^{att}	p^{fill}	precision	recall	F1
Push	10%	5%	0.7339	1.0000	0.8465
	10%	10%	0.8750	1.0000	0.9333
	20%	5%	0.7249	0.9940	0.8384
	20%	10%	0.7915	1.0000	0.8836
Nuke	10%	5%	0.6947	1.0000	0.8198
	10%	10%	0.8667	1.0000	0.9286
	20%	5%	0.7523	1.0000	0.8586
	20%	10%	0.8146	1.0000	0.8978

表 4 均值攻击下 SATPR 算法的查准率,查全率和 F1 度量

	p^{att}	p^{fill}	precision	recall	F1
Push	10%	5%	0.7398	1.0000	0.8505
	10%	10%	0.9891	1.0000	0.9945
	20%	5%	0.8608	1.0000	0.9252
	20%	10%	0.9882	1.0000	0.9940
Nuke	10%	5%	0.7109	1.0000	0.8311
	10%	10%	0.9785	1.0000	0.9891
	20%	5%	0.8564	1.0000	0.9227
	20%	10%	0.9882	1.0000	0.9940

表 5 流行攻击下 SATPR_NoAttr 算法的查准率,查全率和 F1 度量

	p^{att}	p^{fill}	precision	recall	F1
Push	10%	5%	0.7583	1.0000	0.8626
	10%	10%	0.8585	1.0000	0.9239
	20%	5%	0.7511	0.9940	0.8557
	20%	10%	0.8029	1.0000	0.8907
Nuke	10%	5%	0.8053	1.0000	0.8922
	10%	10%	0.8750	1.0000	0.9333
	20%	5%	0.7746	0.9880	0.8684
	20%	10%	0.7877	1.0000	0.8813

表 6 流行攻击下 SATPR 算法的查准率,查全率和 F1 度量

	p^{att}	p^{fill}	precision	recall	F1
Push	10%	5%	0.8273	1.0000	0.9055
	10%	10%	0.9891	1.0000	0.9945
	20%	5%	0.8883	1.0000	0.9408
	20%	10%	0.9940	1.0000	0.9970
Nuke	10%	5%	0.8426	1.0000	0.9146
	10%	10%	0.9891	1.0000	0.9945
	20%	5%	0.9076	1.0000	0.9516
	20%	10%	0.9940	1.0000	0.9970

由实验结果可知，攻击强度与填充率越大，托攻击用户越易于检测，反之将会出现一定的误判。究其原因，由于托攻击通常是批量注入的，当攻击强度较高时，这种隐含的群体性特征通常会有助于连带找出其余的托攻击用户。从填充率来看，当填充率较高时，托攻击用户将会接近更多的正常用户，相比之下也越容易暴露出托攻击用户并非基于兴趣偏好打分而产生的不自然性；而填充率较低时，尚未能根据有限的评分表露出用户偏好，此时，正常用户与托攻击用户的评分概貌差别不大，因而往往会导致误判。

在查全率方面，SATPR_NoAttr 算法在大部分攻击模式下基本达到了 100%，而 SATPR 算法在各种攻击下的查全率均完全达到了 100%。但在查准率方面，二者均未能达到 100%。这说明两种算法基本上都可以将托攻击用户全部查找出来，但同时会将一些正常用户误判为托攻击用户。尽管如此，SATPR 算法在各种攻击下的检测精度均优于 SATPR_NoAttr 算法。在 F1 度量方面，SATPR_NoAttr 算法仅在 90%左右，而 SATPR 算法在一些攻击模式下一度接近 100%，这表示几乎没有遗漏也没有误判。实验结果验证了加入属性信息的有效性。

实验 2 SATPR 算法评分预测实验

在本实验中，分别比较了三种推荐算法在托攻击干扰下的评分预测精度：第一种算法是常见的矩阵分解算法[2]，记作 MF；第二种算法是没有采取抗托攻击措施的 SimMF 算法；第三种算法是 SATPR 算法，这里在评分预测部分采用了 SimMF 算法，是 AFMCSN 托攻击检测模型与 SimMF 算法的结合。实验结果

如表 7~12 所示。

表 7 随机攻击下的 MAE 值

	p^{att}	p^{fill}	MF	SimMF	SATPR
Push	10%	5%	0.6178	0.5829	0.4735
	10%	10%	0.6050	0.5740	0.3139
	20%	5%	0.6268	0.5840	0.4736
	20%	10%	0.6067	0.5757	0.3347
Nuke	10%	5%	0.6165	0.5751	0.4701
	10%	10%	0.6001	0.5707	0.3362
	20%	5%	0.6296	0.5864	0.4831
	20%	10%	0.6045	0.5757	0.3301

表 8 随机攻击下的 RMSE 值

	p^{att}	p^{fill}	MF	SimMF	SATPR
Push	10%	5%	0.7984	0.7276	0.6388
	10%	10%	0.7862	0.7213	0.4240
	20%	5%	0.8216	0.7349	0.6584
	20%	10%	0.7871	0.7243	0.5412
Nuke	10%	5%	0.8051	0.7293	0.6558
	10%	10%	0.7826	0.7197	0.5084
	20%	5%	0.8244	0.7398	0.6853
	20%	10%	0.7919	0.7313	0.5154

表 9 均值攻击下的 MAE 值

	p^{att}	p^{fill}	MF	SimMF	SATPR
Push	10%	5%	0.6142	0.5810	0.3110
	10%	10%	0.6050	0.5773	0.4565
	20%	5%	0.6199	0.5862	0.4668
	20%	10%	0.6171	0.5833	0.3102
Nuke	10%	5%	0.6096	0.5760	0.4828
	10%	10%	0.5967	0.5672	0.3148
	20%	5%	0.6006	0.5715	0.4897
	20%	10%	0.6150	0.5821	0.3397

表 10 均值攻击下的 RMSE 值

	p^{att}	p^{fill}	MF	SimMF	SATPR
Push	10%	5%	0.8017	0.7313	0.4194
	10%	10%	0.7837	0.7268	0.6063
	20%	5%	0.8129	0.7427	0.6491
	20%	10%	0.8067	0.7412	0.4387
Nuke	10%	5%	0.7947	0.7268	0.6530
	10%	10%	0.7807	0.7129	0.4268
	20%	5%	0.7890	0.7283	0.7431
	20%	10%	0.8033	0.7385	0.5209

表 11 流行攻击下的 MAE 值

	p^{att}	p^{fill}	MF	SimMF	SATPR
Push	10%	5%	0.6038	0.5724	0.4370
	10%	10%	0.6150	0.5892	0.3162
	20%	5%	0.6206	0.5828	0.4366
	20%	10%	0.6124	0.5841	0.2921
Nuke	10%	5%	0.6009	0.5737	0.4359
	10%	10%	0.6032	0.5780	0.3091
	20%	5%	0.6247	0.5857	0.4517
	20%	10%	0.6184	0.5881	0.2930

表 12 流行攻击下的 RMSE 值

	p^{att}	p^{fill}	MF	SimMF	SATPR
Push	10%	5%	0.7898	0.7171	0.5865
	10%	10%	0.8086	0.7430	0.4274
	20%	5%	0.8178	0.7367	0.6428
	20%	10%	0.7889	0.7274	0.4092
Nuke	10%	5%	0.7803	0.7250	0.5917
	10%	10%	0.7897	0.7352	0.4204
	20%	5%	0.8188	0.7412	0.6779
	20%	10%	0.8138	0.7455	0.4138

由实验结果可知，注入攻击概貌后会降低推荐精度，在相同的填充率下，攻击强度增大会导致 MAE 与 RMSE 值变大。然而，在很多攻击模式下，相同的攻击强度，填充率越高，MAE 与 RMSE 值反而会变小。由于填充项是攻击者为了伪装成正常用户而注入的，因此填充项是符合原始数据分布规律的，更多的填充项也意味着更多的正常数据，反而提升了数据集的鲁棒性。但对攻击者而言，若填充项过少，又很难影响到目标用户的推荐结果，从而无法达到攻击意图。为应对精心设计的托攻击注入模式，对应的抗托攻击策略也同样需要仔细分析。

在三种推荐算法中，SimMF 算法的性能要优于基本的矩阵分解算法，可见 SimMF 算法充分利用了相似度进行加权，在一定程度上降低了推攻击的干扰。然而，SATPR 算法取得了最优的性能，在各种攻击模式下，均得到了最小的 MAE 与 RMSE 值，这证明了 SATPR 算法在托攻击下的鲁棒性。

4 结束语

本文针对个性化推荐系统面临的托攻击问题，从结构化噪声矩阵补全角度出发，将托攻击用户评分建模为干扰了自然状态下评分矩阵近似低秩性的结构化行噪声，并利用$L_{2,1}$范数对这些托攻击评分进行解析。此外，本文在攻击检测过程中融入用户与物品的属性特征信息，提出了属性优化结构化噪声矩阵补全模型，提高了攻击检测精度。最后，本文基于所提出的 AFMCSN 托攻击检测模型，改进了传统的不具备抗托攻击能力的推荐算法，提出一种基于属性优化矩阵补全的抗托攻击个性

化推荐算法。实验结果证明，在托攻击干扰下，SATPR 算法依旧可以产生鲁棒的评分预测结果，在推荐系统的实际应用中具有现实意义。

参考文献：

[1] Martin F. J, Donaldson J, Ashenfelter A, et al. The big promise of recommender systems [J]. AI Magazine. 2011, 32 (3): 19-27.

[2] Koren Y, Bell R, and Volinsky C. Matrix factorization techniques for recommender systems [J]. Computer. 2009, 42 (8): 30-37.

[3] Gunes I, Kaleli C, Bilge A, et al. Shilling attacks against recommender systems: a comprehensive survey [J]. Articial Intelligence Review. 2014, 42 (4): 767-799.

[4] Zhang Qiang, Luo Yuan, Weng Chuliang, et al. A trust-based detecting mechanism against profile injection attacks in recommender systems [C]// Proc of the 3rd IEEE International Conference on Secure Software Integration And Reliability Improvement. 2009: 59–64.

[5] Bryan K, O'Mahony MP, Cunningham P. Unsupervised retrieval of attack profiles in collaborative recommender systems [C]// Proc of the 2nd ACM International Conference on Recommender System. 2008: 155–162.

[6] Li Wentao, Gao Min, Li Hua, et al. Shilling attack detection in recommender systems via selecting patterns analysis [J]. IEICE Trans on Information & Systems, 2016, E99-D (10): 2600-2611.

[7] Deng Zijun, Zhang Fei, Sandra P S, et al. Shilling attack detection in collaborative filtering recommender system by PCA detection and perturbation [C]// Proc of International Conference on Wavelet Analysis and Pattern Recognition. 2016: 213-218.

[8] 陈蕾，杨庚，陈正宇，等. 基于结构化噪声矩阵补全的 Web 服务 QoS 预测 [J]. 通信学报. 2015, 36 (6): 49-59.

[9] Xiao Fu, Sha Chaoheng, Chen Lei, et al. Noise-tolerant Localization From Incomplete Range Measurements for Wireless Sensor Networks [C]// Proc of INFOCOM. 2015: 2794-2802.

[10] Zhang Zhao, Li Fanzhang, Zhao Mingbo, et al. Robust Neighborhood Preserving Projection by Nuclear//L2, 1-Norm Regularization for Image Feature Extraction [J]. IEEE Trans on Image Processing, 2017, 26 (4): 1607-1622.

[11] 汤镇宇，孟凡荣，王志晓. 基于稀疏表示的快速 $L2$-范数人脸识别方法 [J]. 计算机应用研究. 2016, 33 (9): 2831-2836.

[12] Srebroand N, Jaakkola T. Weighted low-rank approximations [C]// Proc of International Conference on Machine Learning, 2003: 720-727.

[13] Candès E J, Recht B. Exact matrix completion via convex optimization [J]. Foundations of Computational Mathematics, 2009, 9 (6): 717-772.

[14] Cai Xiao, Nie Feiping, Huang Heng, et al. Multi-Class L2, 1-norm support vector machine [C]// Proc of the 11th IEEE International Conference on Data Mining. 2011: 91-100.

[15] Xu Miao, Jin Rong, Zhou Zhihua. Speedup matrix completion with side information: Application to multi-label learning [C]// Advances in Neural Information Processing Systems. 2013: 2301-2309.

[16] Beck A, Tetruashvili L. On the convergence of block coordinate descent methods [J]. SIAM Journal on Optimization, 2013, 23 (4): 2037-2060.

[17] Combettes P L, Wajs V R. Signal recovery by proximal forward-backward splitting [J]. SIAM Journal on Multiscale Modeling & Simulation, 2005, 4 (4): 1168-1200.

[18] Ng A Y, Jordan M I, Weiss Y. On spectral clustering: analysis and an algorithm [C]// Proc of International Conference on Neural Information Processing Systems: Natural and Synthetic. Cambridge: MIT Press, 2001: 849-856.

[19] Shi Chuan, Huang Yue, Philip S. Yu, et al. HeteSim: A General Framework for Relevance Measure in Heterogeneous Networks [J]. IEEE Trans on Knowledge and Data Engineering. 2014, 26 (10): 2479 – 2492.

[20] Shi Chuan, Liu Jian, Zhuang Fuzhen, et al. Integrating Heterogeneous Information via Flexible Regularization Framework for Recommendation [J]. Knowledge and Information Systems. 2016, 49 (3): 835–859.

[21] 李聪，骆志刚，石金龙. 一种探测推荐系统托攻击的无监督算法 [J]. 自动化学报, 2011, 37 (2): 160-167.